

João Vitor Montanari

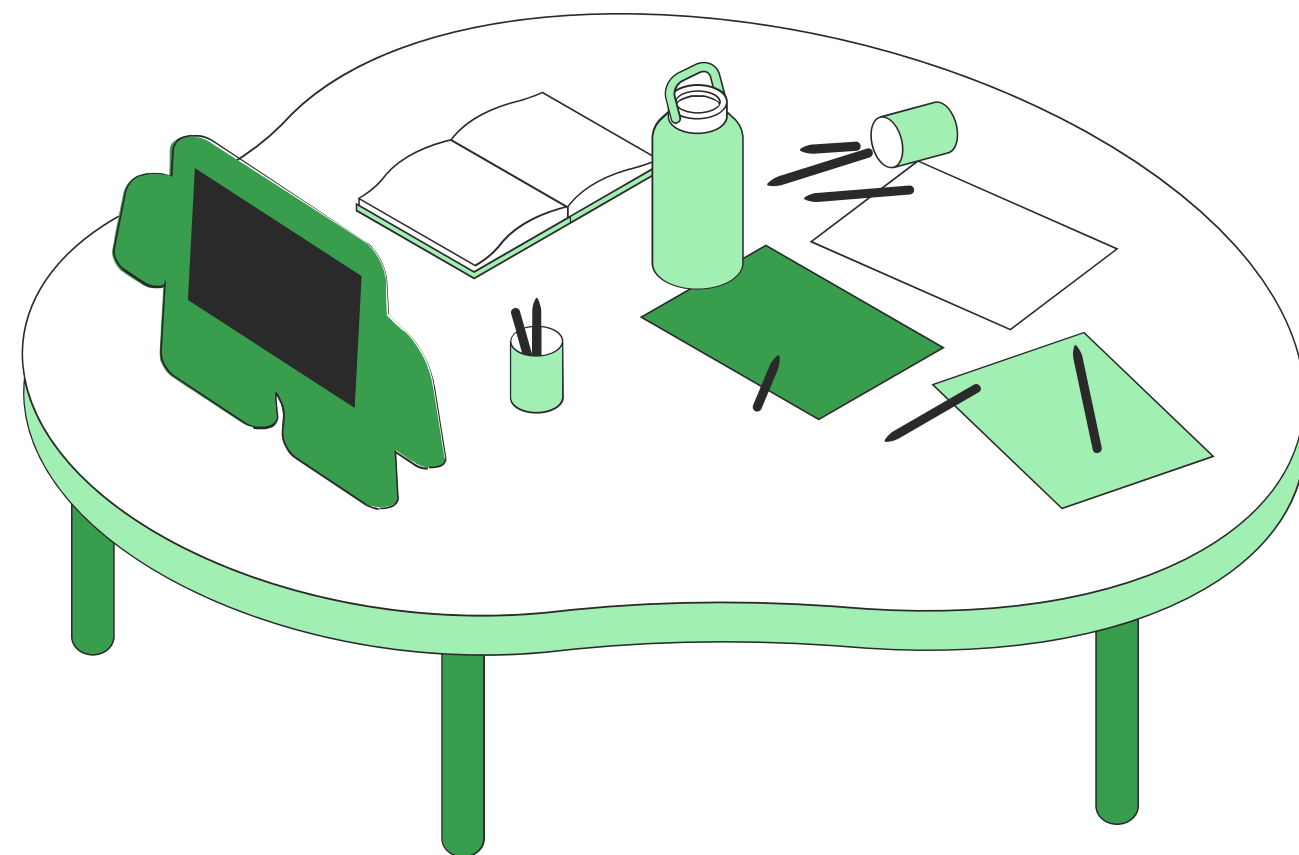
# PROLOG

Resumo de aulas



# Sumário

## Summary



Introdução

---

História

---

Tipos de Dados

---

Fatos

---

Regras

---

Operadores de Controle

# O que é:

O Prolog se trata de uma linguagem declarativa, logo a mesma não demonstra uma maneira de chegar até a estipulação de um determinado problema através de um passo a passo como as linguagens imperativas e procedurais, ela se limita a ideia de fornecer uma descrição do problema que foi apresentado a ela. Assim está usa uma coleção de base de dados, de fatos e de relações lógicas (regras).

Um programa escrito em Prolog roda a partir de consultas (queries) formuladas pelo próprio usuário, assim utilizando da base de dados (os fatos), as regras relacionais e o mecanismo de unificação para poder chegar na solução.

A execução do Prolog é efetivamente a prova de um teorema por resolução de primeira ordem. Alguns de seus conceitos fundamentais são unificação, recursão e backtracking.



# História

A linguagem de programação Prolog nasceu de um projeto que não tinha por foco a implementação de uma linguagem de programação, mas o processamento de linguagens naturais.

Na Universidade de Marselha, Alain Colmerauer e Robert Pasero trabalhavam na parte de linguagem natural e Jean Trudel e Philippe Roussel trabalhavam na parte de dedução do projeto.

O projeto resultou em uma versão preliminar da linguagem Prolog em fins de 1971 sendo que a versão definitiva apareceu em fins de 1972.



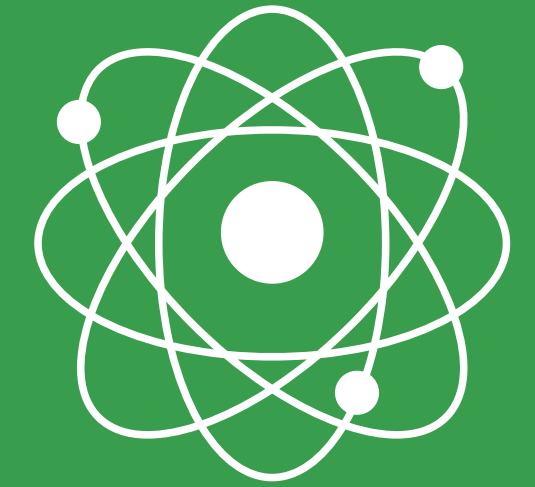
# Tipos de dados:

Prolog não emprega tipos de dados do mesmo modo que as linguagens de programação mais comuns normalmente fazem. Todos os dados são tratados como sendo de um único tipo, **Termo**, cuja natureza depende da forma como esse termo foi declarado. Ou seja, os elementos léxicos utilizados na sua declaração determinam se esse termo será um número, um texto, uma variável, uma estrutura complexa e assim por diante.





# Átomos:



As constantes de texto são chamadas de átomos. Um átomo é uma sequência constituída de letras, números e underscore, mas iniciando com uma letra minúscula. Se um átomo não alfanumérico é necessário, pode-se usar qualquer sequência entre aspas simples.

Exemplo:

Começando com letra minúscula:

pedro henrique\_vi

Como uma sequência de caracteres entre aspas simples:

'Hello world' 'como assim?'

# Números

Um número é uma sequência de dígitos, permitindo também os sinais de . (para números reais), - (número negativo) e e (notação científica). Algumas implementações do Prolog não fazem distinção entre inteiros e números reais.

Exemplo:

3   3.14   -3   3.14e

**Ponto flutuante:**

●

**Negativo:**

—

**Notação Científica:**

e



# Fatos

Programar em Prolog é bem diferente de programar em uma linguagem procedimental. Em Prolog se fornece fatos e regras para uma base de dados, então se executam consultas ou (queries) a essa base de dados. A unidade básica do Prolog é o predicado, que é postulado verdadeiro. Um predicado consiste de uma cabeça e um número de argumentos.

Exemplo:

`gato(tom)`

→ Isso informa à base de dados o fato que 'tom' é um 'gato'. Formalmente, 'gato' é a cabeça e 'tom' é o único argumento do predicado. Veja alguns exemplos de consultas que podem ser feitas ao interpretador Prolog baseado nesse fato:

**Tom é um gato?**

`?- gato(tom).`

`yes.`

**que coisas (conhecidas) são gatos?**

`?- gato(X).`

`X = tom;`

`yes.`



# Regras



O segundo tipo de predicado no Prolog é a regra, também chamada de "cláusula".  
Um exemplo de uma regra é:

```
luz(acesa) :- interruptor(ligado)
```

O ":-" significa "se"; essa regra significa que luz(acesa) é verdadeiro se interruptor(ligado) é verdadeiro.

Regras podem também fazer uso de variáveis, como por exemplo:

```
avo(X,Z) :- pai(X,Y), pai(Y,Z).
```



# Operadores de controle

## Backtracking

É um procedimento dentro da linguagem Prolog. Uma busca inicial em um programa nesta linguagem segue o padrão Busca em profundidade, ou seja, a árvore é percorrida sistematicamente de cima para baixo e da esquerda para direita. Quando essa pesquisa falha, ou é encontrado um nó terminal da árvore, entra em funcionamento o mecanismo de backtracking. Esse procedimento faz com que o sistema retorne pelo mesmo caminho percorrido com a finalidade de encontrar soluções alternativas.

Exemplo:

Considerando uma base de dados família, fazemos a seguinte consulta:

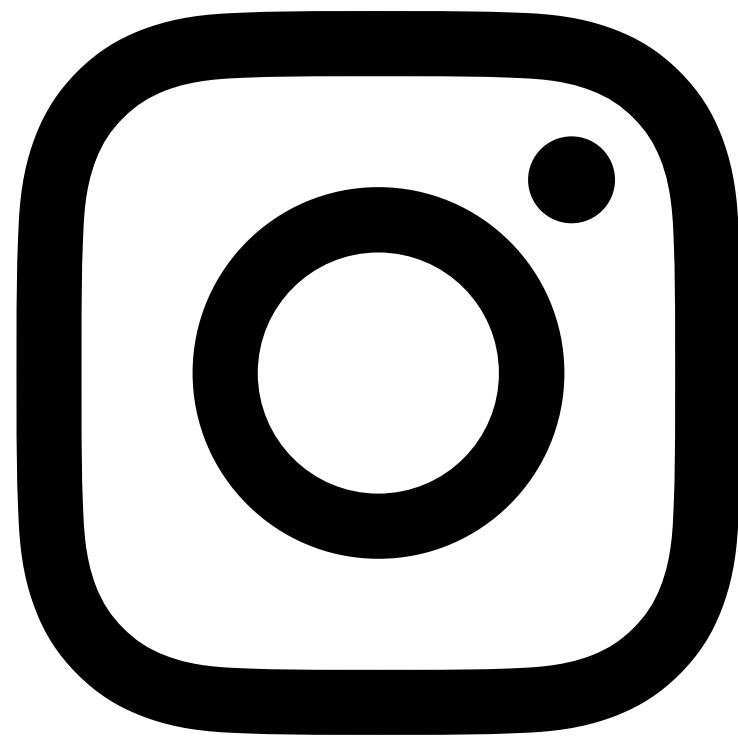
```
?- pai(roberto,X), mae(vera,X)
```

O compilador tenta satisfazer o primeiro objetivo. Quando conseguir, tenta satisfazer o segundo. Caso não consiga, ele retorna ao ponto onde encontrou a solução para o primeiro objetivo (backtracking).

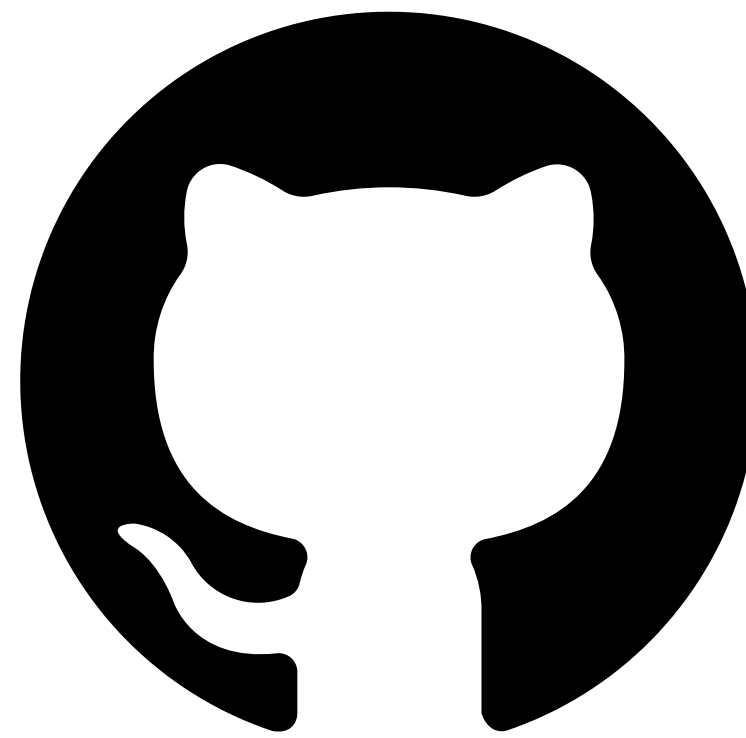


# Me siga nas redes sociais:

Espero que tenha gostado!  
Aproveite e me siga nas redes sociais :)



@joaomontanari26



joao-montanari



João V. Montanari